Fuzzy Transfer Learning: Methodology and Application

Corresponding Author: Jethro Shell[1,*], Simon Coupland[1,**]

De Montfort University,United Kingdom

Abstract

Producing a methodology that is able to predict output using a model is a well studied area in Computational Intelligence (CI). However, a number of real-world applications require a model but have little or no data available of the specific environment. Predominantly, standard machine learning approaches focus on a need for training data for such models to come from the same domain as the target task. Such restrictions can severely reduce the data acquisition making it extremely costly, or in certain situations, impossible. This impedes the ability of these approaches to model such environments. It is on this particular problem that this paper is focussed.

In this paper two concepts, Transfer Learning (TL) and Fuzzy Logic (FL) are combined in a framework, Fuzzy Transfer Learning (FuzzyTL), to address the problem of learning tasks that have no prior direct contextual knowledge. Through the use of a FL based learning method, uncertainty that is evident in dynamic environments is represented. By applying a TL approach through the combining of labelled data from a contextually related source task, and little or no unlabelled data from a target task, the framework is shown to be able to accomplish predictive tasks using models learned from contextually different data.

Keywords: fuzzy logic, transfer learning, context, online adaptation, sensor networks, intelligent environments
94D05 MSC: ,
68T05 MSC:

1. Introduction

The availability of information impacts on the understanding of a problem. Further to this, a lack of information reduces the ability to understand a problem. Differences in information and understanding about one problem domain and a second similar domain can be defined as being contained within a *knowledge gap*. This paper presents a novel composition of methods that use a combination of inductive and deductive learning mechanisms that draw inspiration from human approaches to problem solving, to bridge the knowledge gap. Two concepts, Transfer Learning (TL), a methodology that allows information gained in different contextual situations to assist new learning tasks[2], and Fuzzy Logic (FL), an approach to capture imprecision and uncertainty, are brought together in a novel framework to address the problem of learning tasks that have no prior direct contextual knowledge.

Real-world applications often consist of many unknowns increasing the knowledge gap. To predict or classify based on the information gathered from these applications can be difficult. Standard machine learning approaches require that there is a form of training data. Predominantly such training data has to come from the same domain. Some applications make the procurement of *a priori* labelled training data demanding, or in some cases, not possible at all. For example, to measure certain physical areas such as remote forest locations, impromptu set ups such as disaster zones, or small user groups that have very defined requirements such as disabled users.

[*] +44 (0)116 255 1551
[**] +44 (0)116 207 8419
Email addresses: jethros@dmu.ac.uk (Corresponding Author: Jethro Shell), simonc@dmu.ac.uk (Simon Coupland)
[1] The Centre for Computational Intelligence, De Montfort University, The Gateway, Leicester, LE1 9BH, United Kingdom
[2] In this research the term *task* is referred to as any action the learning method is required to accomplish

The procurement of training data produces an interesting problem. If there is a requirement to classify or predict the output from such environments, *how can a model be produced?* The examples given previously present situations where labelled data from the same distribution is costly. The lack of information within the same problem domain can cause standard supervised learning to be ineffective. The process of labelling data can be costly. For example, there may be few images that are labelled from a given feature space [1]. Recognising that supervised methods require that training data is supplied from the same domain, Semi-Supervised Learning (SSL) was introduced to approach this issue [2]. SSL sits in between supervised and unsupervised learning. Unlike supervised learning, where the goal is to learn a mapping from x to y, given a training set of pairs (x_i, y_i), SSL is supplied with unlabelled data. $y_i \in Y$ is referred to as the labels of x_i. Typically the focus of unsupervised learning is to find structure in the unlabelled data, $X = (x_1, \ldots, x_n)$ where $x_i \in X$ for all of $i \in n$ [3]. Additionally, large quantities of unlabelled data may not be available. The initial reduced quantity of unlabelled domain data also renders the use of SSL and unsupervised less effective.

The contexts discussed, however, can be related to other implementations which may contain previously discovered knowledge. The transferral of knowledge from one context to another is in keeping with the concept of a more humanist style of learning, to reuse and repurpose information. Within the study of human learning, ordinary learning is viewed as being ordinary when it is within the same context (a student may solve similar problems that are at the end of a chapter that have appeared previously), whereas TL occurs outside of a single context (problems are solved when they occur mixed with others at the end of the course) [4]. Studies have shown that humans often draw upon more than just training data for generalisation [5]. In recent years there has been significant quantities of research in the area of TL and its application to real-world problems in the area of Computational Intelligence (CI) [6, 7, 8, 9]. TL can be broadly defined as a learning technique that uses knowledge from a source domain to increase the performance of learning within the target task domain. The methodology allows the domains, tasks and distributions used within the training and testing to be different. The research within this paper presents a novel use of a TL method to model scenarios where little or no information is initially known.

There is a strong relationship between context and uncertainty. As individuals endeavour to learn a new task they often afford uncertainty to it. There is a clear codependency on the level of certainty in any learning activity and the amount of information that is available. Problems with little information can have a high degree of uncertainty [10]. Dynamic applications such as Intelligent Environments (IEs) can exhibit this uncertainty in the sensors that are used and the decision structures that are applied. The incorporation of a FL system is proposed to assist in the modelling of environments in the presence of uncertainty and vagueness. The use of FL allows for the incorporation of approximation and a greater expressiveness of the uncertainty within the data [11].

To summarise, this paper presents a novel framework, Fuzzy Transfer Learning (FuzzyTL), that uses the methods within FL and TL to bridge the *knowledge gap* between the learning process of one context to another. Whilst the abilities of the framework have been shown to be applied to predictive tasks (as illustrated in Section **??**), there is a belief that the generic nature of the framework allows it to be applied to problem spaces beyond these confines. The paper illustrates that the use of FuzzyTL can outperform other state of the art methodologies in the prediction of data for contexts that have little or no apriori knowledge.

The structure of the paper is as follows. The next section, Section 2 gives an overview of the current research in this area. Section 3 describes the methodology used within the FuzzyTL framework. The experiments and results carried out to analyse the proposed methodology are discussed in Section 4. The final two sections conclude the paper and outline future work.

2. Background

2.1. Context

The defining of context is broad and far reaching. The study of context is a multi-disciplinary pursuit, ranging from psychology and linguistics, to computing (especially within Computational Intelligence).

The concept of context plays an important role in both FL and TL. There is, however, no single consensus of how context should be defined. The structure of the FuzzyTL framework has foundations in the notion of context. TL has the ability to use information from one domain to close the information gap in a learning process from a differing but similar domain. The domains can be defined as contexts. To analyse the contexts, a valid definition of a context must be put forward.

There has been considerable work surrounding context and computing. Dey [12] puts forward a definition of context:

> "Context is any information that can be used to characterise the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves."

The definition by Dey relates to Dourish's representational view of context [13]. Dourish uses the representational nature of software systems to represent and encode context. This view has similarities to definitions by Schilit et al. [14] and Jang [15] as each describe context through its relationship to information which is formed or expressed, to varying degrees of abstraction. Dourish asserts four assumptions regarding context that are based upon these types of definition. They are:

1. *Context is a form of information.* It is something that can be known and therefore represented.
2. *Context is delineable.* For an application, or set of applications, the context of the activities which the application supports can be defined.
3. *Context is stable.* Variation may occur within elements of the application from application to application, however they do not vary from instance to instance.
4. *Context and activity are separable.* Context describes the features of an environment that an activity occurs within.

2.1.1. Defining Context

To consolidate the varying descriptions of context, a high level, abstract definition is given, and subsequently used within this paper. Taking influence from the work of Dey [12], Dourish [13] and Bettini et al. [16], context is defined as:

1. **Information:** Each context consists of definable variables that are relevant and measurable.
2. **Behaviour:** The context embodies an entity, application, service or group thereof that is affected by the behaviour of the associated information.
3. **Variation:** Differences within the structure of the variables can occur between context to context, but not from instance to instance within a context itself. This would be defined as a new context.

This can be represented formally as:

$$\alpha_1 = \{a_i, b, c\}^j \ldots \alpha_2 = \{a, b, d\}^k$$

where α is a context containing multiple variables (a, b, c, or d) and varying quantities of those variables j and k. Each context is definable as it contains variables that are measurable. The two contexts show variation as they contain differing variable structures. α contains a variable a with a differing structure to α_2. This variation does not occur within the context itself, however.

Context is directly linked to the measurement of the real-world. The measuring of any application must take into account the context in which it exists. As both the real-world and subsequently the context it embodies are evaluated, the uncertainty and vagueness that are contained become more apparent. There is a need to capture and effectively represent this uncertainty and vagueness.

2.2. Uncertainty

Much of science requires the pursuit of precision and exactness. However, humans live in a world that is formed by imprecision, vagueness and uncertainty. Real world applications are particularly at the mercy of this world. As people endeavour to measure the world, imprecision emerges. The cost associated with the pursuit of increasing precision rises in equal measure. As an example, parking a car is a simple task as it only requires that the final placement of the vehicle is imprecise. Generally, parking spaces allow for a large margin of error. Decreasing the error margin from many centimetres to only a few millimetres, and so increasing the precision, would drastically increase the cost associated in terms of execution [17]. Similarly, uncertainty is codependent on the quantity of information that is

available. As more information about a problem is acquired, individuals become more certain about its formulation and solution. Problems with less information have a higher degree of uncertainty [10].

By its very nature, uncertainty increases with a lack of knowledge. Uncertainty can be considered as existing in the *knowledge gap*. A knowledge gap can be broadly defined as the level of understanding that is exhibited based on the information that is known, compared to an optimum level of understanding. Using the previous car parking example, the optimum level of understanding may be considered to be how a driving instructor will park a vehicle having full vision of the parking bay. Comparably, a student may have a reduced level of understanding. In this scenario, information may be limited, conflicting and vague because of limited driving experience and vehicles partially obscuring the view of the parking bay. This alters the learning and understanding of the task, manifesting itself as uncertainty. The difference in the understanding of the task that the instructor and the student exhibit is expressed by the knowledge gap.

To tackle the uncertainty and vagueness that is exhibited within the real-world, the framework proposed in this paper is based upon the use of FL. FL is used to capture the uncertainty and vagueness that exits in data. The FL elements of the framework are brought together in a Fuzzy Inference System (FIS). A strength of the FIS is the ability to handle linguistic concepts and perform non-linear mapping between inputs and outputs [18]. FIS takes crisp inputs and maps them to crisp outputs. A FIS principally contains four components: fuzzy rules, a fuzzifier, an inference engine, and a defuzzifier [19]. For a full description of a FIS, see [10] and [19]. Various automated learning methods can be used to generate the main elements of a FIS, namely the fuzzy sets and the rulebase. Each learning method approaches one or a number of these areas of the FIS. The framework proposed in this paper uses a Ad-Hoc Data Driven Learning (ADDL) method.

2.3. Wang-Mendel Methodology

The FuzzyTL framework identifies the structure of the FIS through the use of an automated learning process. Numerical source data is used to form fuzzy rules and fuzzy sets. The process is based on an algorithm proposed by Wang and Mendel [20].

The basis of the Wang-Mendel (WM) process is the formation of fuzzy sets and fuzzy rules that constitute the main components of the FIS. The main element of the method is the generation of fuzzy rules from numerical pairs which in turn are formed into a rulebase. The approach is a generalised one and is in keeping with the requirements of the TL structure used in the FuzzyTL framework. Wang and Mendel[20] proposed a multi-step procedure to produce the fuzzy rules and fuzzy sets. This is described in Figure 1.

The WM initially divides each domain interval into fuzzy regions, each containing the membership functions for that input or output. In order to automate this step, the domain is equally divided based upon the minimum and maximum values of the interval and the defined number of regions. Rules are also created by generating membership values from each data point. To remove conflicts and reduce the rulebase to a manageable size, the WM algorithm retains those rules with the highest membership in the same group. The final stage is to map the input to the output values. This is achieved through a defuzzification process.

Although a number of data driven learning methods were considered for the FuzzyTL framework, the simplistic and easy approach to the implementation of the WM method allows for a quick adoption of the framework. Equally, the ability to modify its components are fundamental to the FuzzyTL frameworks adaptive approach. The swiftness in the execution within the early stages of the preliminary fuzzy modelling process allows for subsequent adaptation of the model by other methods [21]. Its level of generality and maturity have allowed it to be applied to a large number of applications, which is applicable to the FuzzyTL approach.

2.4. Transfer Learning

The motivation of transfer learning is to improve the learning in a target domain by acquiring information from a differing but related domain. Traditional machine learning strategies work under a number of assumptions. Mihalkova et al. [22] propose that the learning of each new task begins from scratch. Many machine learning techniques also require that training and testing data is sourced from the same feature space. This can be difficult, or in some cases impossible. TL offers the ability to apply previously acquired knowledge to areas where little or no information is available, improving the learning. TL has been applied to varying domains: activity recognition [8], eye tracking [23], gaming [24] and image classification [25] to name but a few. A major motivation behind the FuzzyTL framework comes from environments that lack any prior knowledge in the form of labelled training data.

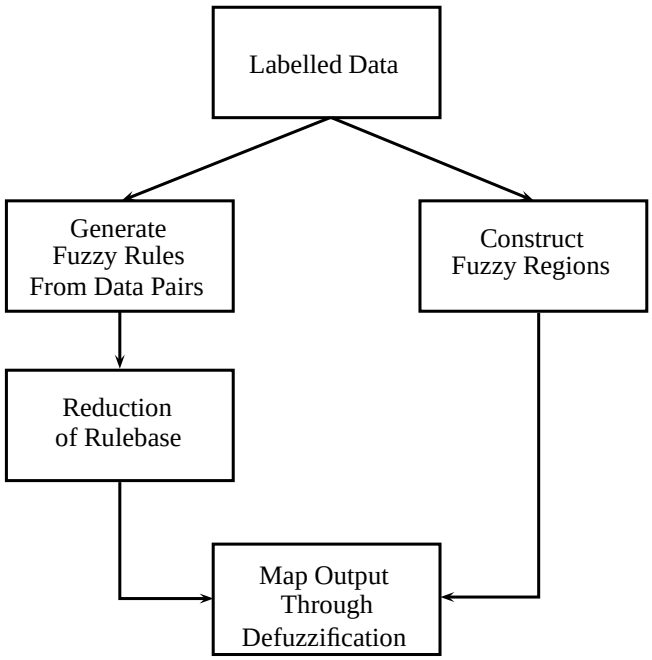


Figure 1: Multi-step Wang-Mendel [20] Process.

2.4.1. Measures, Definition and Foundations

Transfer learning contains two principle elements, a *Domain* and a *Task*. According to Pan and Yang [26], a *Domain* can be defined as consisting of two components: a feature space x and a marginal probability distribution $P(x)$ where $X = \{x_1, \ldots, x_n\} \in X$. A *Task* consists of a label space Y and a predictive function $f(.)$. The predictive function can be learned from the training data which is constructed as data pairs $\{x_i, y_i\}$ where $x_i \in X$ and $y_i \in Y$. The *source domain* can be defined as $\mathcal{D}_s = \{(x_{s_1}, y_{s_1}), \ldots, (x_{s_n}, y_{s_n})\}$ where $x_s \in X_s$ is the data point and $y_s \in Y_s$ is the corresponding label.

Based on these definitions transfer learning can be defined as:

> Given a source domain $\mathcal{D}_s$ and a learning task $\mathcal{T}_s$, a target domain $\mathcal{D}_t$ and a learning task $\mathcal{T}_t$, transfer learning aims to improve the learning of a new task $\mathcal{T}_t$ through the transfer of knowledge from a related task $\mathcal{T}_s$ [27] by the learning of the predictive function in the target domain $\mathcal{D}_t$, where $\mathcal{D}_s \neq \mathcal{D}_t$ or $\mathcal{T}_s \neq \mathcal{T}_t$ [26].

There are a number of differing types of TL. The FuzzyTL framework uses an Inductive Informed Unsupervised Transfer Learning method.

2.4.2. Transductive Informed Unsupervised Transfer Learning

Transductive transfer learning is derived from classical transductive learning. The process of transductive learning uses both a labelled training set of data combined with an auxiliary unlabelled set [**?**]. This is in contrast to inductive learning which is the process of acquiring knowledge by drawing inductive inferences from teacher or environment-provided facts [**?**]. In the context of TL, this is where labelled data exists in the training data and within the testing data, but the test data is hidden from the process. As with standard machine learning approaches, there are number of variations based on the availability of training data. Within this paper, we will use the terminology of Informed Unsupervised (IU), Informed Supervised (IS), Uninformed Unsupervised (UU) and Uninformed Supervised (US).

The term IU transfer learning was introduced by Cook, Feuz and Krishnan [28]. Deviating from the standard terms of *supervised* and *unsupervised* learning, they introduced the use of *informed* and *uninformed* which are applied to the availability of labelled data in the source and target areas. IS transfer learning implies that labelled data is available in both the target and source domains. Labelled data in the target domain is required to induce an objective predictive

model. In transductive transfer learning both $\mathcal{D}_s = \{(x_s, y_s)\}$ and $\mathcal{D}_t = \{(x_t)\}$ are known where $\mathcal{D}_s$ is the source data and $\mathcal{D}_t$ is the training data. Additionally there is auxiliary unlabelled data that is not part of the training set [26]. IU transfer learning, however defines that the labelled data is only available in the source domain. By contrast, US learning implies that labelled data is available only in the target domain with UU transfer learning implying that there is no availability of labelled data in either domains.

2.4.3. Limited-Data Transfer Learning Methods

Within TL there are a number of sparse and limited-data methods. In this context, limited-data refers to scenarios with datasets that are a low percentage of the overall quantity. One-Shot learning highlights this approach [29]. The basis of One-Shot Learning can be identified by drawing parallels with the abilities of humans to identify objects under a wide variety of conditions after seeing only a single point. The process acquires knowledge from one setting and uses it in another. The methodology models new classes of objects based only on samples of related or support classes through probability densities.

Larochelle et al. [2] expand the concept of limited data with the introduction of *Zero-data Learning*. Zero-data learning is based on the premise that a model must generalise to classes or tasks where there is no availability of training data, only a description of the data. It is assumed that the situation may occur that no labelled data is available so descriptions are used. There are similarities between the problem proposed within this paper and that approached by Larochelle et al. In [2], the authors assumed that the descriptions that are used within the classification process are predefined. Within *Zero-data Learning* the hierarchical definitions can often come from expert opinion, differing from the automated, data-driven strategy chosen within the FuzzyTL approach.

3. Fuzzy Transfer Learning Methodology

A facet of learning is the ability to transfer information from one context to another. Information gained through learning can be generalised, absorbing inconsistencies and anomalies. The hypothesis of the framework is that what has been learnt can be adapted in order to accomplish the new task, building upon and adapting the previous knowledge. In this section the major elements of the FuzzyTL methodology are outlined.

3.1. Overview

The FuzzyTL methodology is contained within a framework structure. The key components can be seen in Figure 2.

In this structure there are two distinct processes: the transferring of the fuzzy concepts and their relationships, and the adaptation of the fuzzy components using knowledge of the application context.

In the first stage the system uses a source of labelled data to instigate a learning process. The learning process uses this source data to construct a FIS. The structure of the FIS consists of fuzzy sets and fuzzy rules. The FIS is used to capture the knowledge from the source, and transfer it to the target task. This process of transferring information is a fundamental aspect of the FuzzyTL methodology, and highlights the use of an IU [28] TL method.

Unlike IS TL where a quantity of labelled data is required from both the source and the target task, IU TL describes situations where no labelled data is available from the target task. The FuzzyTL methodology captures information from the source task to act as an initial learning point for the target task. This is the basis for the TL process.

The second stage of the framework addresses the adaptation of the FIS. The adaptation process uses knowledge from the unlabelled task dataset coupled with previously learnt information. This process adapts the individual components of the FIS to capture the variations in the data. Alterations and variations from situation to situation, are absorbed through changes made within the domains of the fuzzy sets and adaptations to the rulebase. Using this structure, the FuzzyTL methodology is shown to be able to use the transfer of information to assist in bridging the knowledge gap. Through an online adaptation process, newly accrued information can be absorbed.

To continue the discussion of the novel FuzzyTL methodology, a series of elements need to be predefined. A source domain $\mathcal{D}_s$ can be defined as

$$\mathcal{D}_s = \{(x_n^s, y_m^s)\}_s^P \tag{1}$$

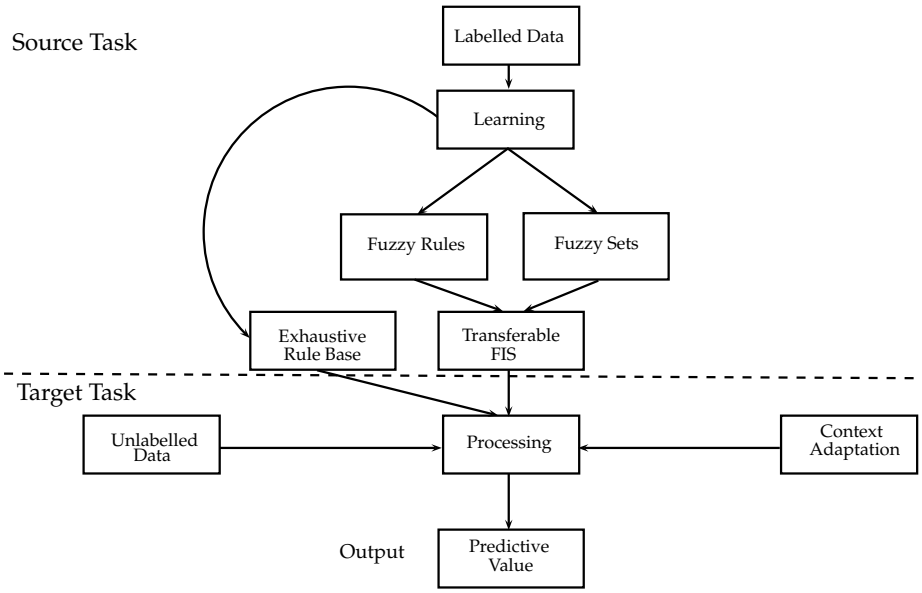


Figure 2: Overview of the Fuzzy Transfer Learning Framework.

where $x \in X$ are data inputs, $y \in Y$ is an output, n is the number of inputs, m is the number of outputs, and P is the number of data tuples within the domain. Equally, the target $\mathcal{D}_t$ and adaptive domains $\mathcal{D}_a$ can be defined in the same way.

Additionally, the domain can be defined through the use of intervals. Intervals are used to represent the domains to allow for the use of interval arithmetic. This has been demonstrated previously within the application of fuzzy temporal relationships [30]. Within this paper, an interval is referred to as a bounded set of real numbers

$$\begin{aligned} A &= [a_L, a_R] = \{a : a_L \leq a \leq a_R, a \in \mathbb{R}\} \\ B &= [b_L, b_R] = \{b : b_L \leq b \leq b_R, b \in \mathbb{R}\} \end{aligned} \tag{2}$$

where a_L and a_R are the left and right limits of the interval A [31]. Two intervals A and B are considered equal if their corresponding endpoints are equal. So, $A = B$ if $a_L = b_L$ and $a_R = b_R$. The intersection of two intervals is empty $A \cap B = \emptyset$, if $a_R < b_L$ or $b_L > a_R$ [32].

Based on this definition and using the interval notation, a source domain interval can be defined as
$\mathcal{D}_s^{\mathcal{I}} = \{[x^s_{1L}, x^s_{1R}], [x^s_{2L}, x^s_{2R}], [{y^s}_L, {y^s}_R]\}$.

A domain will also be defined through its relationship to fuzzy sets. A source domain with n inputs, and m outputs is defined as

$$\mathcal{D}_s = \{{}^f X^s_n, {}^f Y^m\} \tag{3}$$

where ${}^f X^s_n$ is a set of fuzzy input sets, and ${}^f Y^m$ is a set of fuzzy output sets. ${}^f X^s_n$ can be defined as

$${}^f X^s_n = \{vs^{{}^f x^s_n}, s^{{}^f x^s_n}, m^{{}^f x^s_n}, l^{{}^f x^s_n}, vl^{{}^f x^s_n}\} \tag{4}$$

where *vs* to *vl* are the sets *very small*, *small*, *medium*, *large* and *very large* respectively. These sets can be any description that is suitable to the context. Within this paper, the fuzzy sets are constructed as normal, continuous and triangular.

A target domain with l inputs is defined as

$$\mathcal{D}_t = \{{}^f X^t_n\} \tag{5}$$

where ${}^f X^t_n$ is a set of fuzzy input sets.

The rulebases used within the subsequent sections are defined using the same notation. A rulebase that contains n antecedents and m consequent sets is depicted as

$$\mathcal{R} = \{{}^{f}X_{n}^{r}, {}^{f}Y_{m}^{r}\}^{P} \tag{6}$$

where R is the rulebase, X is a data input, Y is the corresponding output, and P is the number of rules.

3.2. Transferring Fuzzy Concepts

The first stage of the FuzzyTL process is the construction of the FIS (Fuzzy Inference System). Fuzzy rules and fuzzy sets are formed via the use of an ADDL process which is calculated from numerical data. The method uses numerical data to form the sets and rules, a procedure based on an algorithm proposed by WM [20].

The basis of the WM process is the formation of fuzzy sets and fuzzy rules from numerical data. The use of the method is not restricted to an individual application domain and has been shown to be applicable to a broad number of implementations [33, 34, 35].

The FuzzyTL framework transfers information from context through the use of a model based on a FIS. The WM algorithm uses numerical data to extract the fuzzy sets and a rulebase that form constituent parts of the FIS.

The FuzzyTL framework follows the standard WM process for the production of the fuzzy sets and fuzzy rules. A full description of the WM process can be found in [20].

A fundamental aspect of the WM process is the reduction of the produced of the fuzzy rulebase. Under the standard WM process, a weighted measure is used to produce a Reduced Rule Base (RRB). This emphasises the rules with the highest antecedent and consequent membership values, reducing the exhaustive rulebase to a reduced set of rules. Previous research has shown that within the exhaustive rulebase there exists further information that can inform both the RRB construction and further enhance the transfer process. The FuzzyTL framework adds to the WM method by supplementing the process with a fuzzy frequency measure. The addition of the fuzzy frequency measure endeavours to remove the possibility of anomalous data influencing the production of fuzzy rules. An in depth explanation of this process can be found in [36].

To capture more of the information contained within the exhaustive rulebase, the FuzzyTL framework uses a frequency value. This value gains more information from the original rulebase by focusing on the number of occurrences of each rule.

3.3. Adaptation Through Learning

As outlined in Section 3.2, the transferral of the FIS embodies the TL component of the FuzzyTL methodology. Using the notation in Section 2.4.1, this can be described as transferring a source domain $\mathcal{D}_s$ to model a predictive function of a target domain $\mathcal{D}_t$. The relationship between $\mathcal{D}_s$ and $\mathcal{D}_t$ influences the output of the model. If there exists some relationship, explicit or implicit, between the two domains this is categorised as being *related*. The nature of the relationship will dictate the necessity for the adaptation of the knowledge contained within the source domain and the learning task. If the domains are equal and the learning tasks are approaching the same problem, no adaptation is required, however, this is rare within real world applications. Separation of the domains results in the need for an adaptation process.

In order for the framework to absorb such changes from the source to the target contexts, the elements of the transferable FIS are adapted. Using the knowledge housed within the exhaustive rulebase, the FIS itself and newly acquired information, changes are made in order for the framework to output the required data. The adaptation process consists of five stages.

3.3.1. External Input Domain Adjustment: Stage One

A knowledge gap can occur during the transfer of learning structures from one contextual domain to another. This can be captured as both differences in the domains themselves, and differences in the learning structure. To absorb such contextual differences in the source and target tasks, the FuzzyTL adapts the minimum and maximum values within the domain. Taking each input instance of the dataset, the framework adjusts the range of the interval according to any difference calculated between the transferable FIS and the new input values.

A source domain $\mathcal{D}_s$ can be represented as data tuples (x_1, x_2, y), where x_1 and x_2 are inputs and y represents the output. A new domain is formed based on $\mathcal{D}_s$ incorporating the alterations made through the adaptation process. This is defined as $\mathcal{D}_a$. $\mathcal{D}_a$ represents missing information between the source and target tasks.

Each data point is analysed to extract information. The input interval is adapted if the value extends beyond the left (x_L) or right (x_R) boundaries. This produces a new set structure. Any adaptation to the domain results in an equal distribution across the sets. This is due to the equal spacing. Extension of the domain requires a simple change to the footprint of each set.

Unevenly distributed membership functions require a scaling function in order to adapt the sets. In the FuzzyTL framework triangular functions are used, however, other functions are applicable. A triangular function can be defined as:

$$A(x) = \begin{cases} (1 - \frac{|x-a|}{s}) & \text{when } a - s \leq x \leq a + s \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

where x is the input value, a is the centre of the function and s is the width. In a similar manner to equally distributed sets, the centre of the function a is used as the anchor point. If the domain is shifted in a negative or positive direction, the sets are moved by the centre points. Each point is moved an equal distance. Any extension or compression of the domain requires that the sets are altered according to the scaling. This process is shown in Figure 3a. Here, three sets are shown in the X universe. The sets shown (*Small*, *Medium* and *Large*) are uneven and have differing footprints. The example shows the domain increased by 30%. If the domain of the target task is contained within the source task, an alternative strategy is required to adapt both set structure and domains.

3.3.2. Internal Input Domain Adjustment: Stage Two

The second adaptation stage also focuses on the input domains. The transferring of source domains can require adaptation to remove the knowledge gap. The knowledge gap can be represented by differences in the domain intervals. In Figure 3b, interval $\mathcal{D}^{\mathcal{I}}_{t_1}$ is shown to be a subset of $\mathcal{D}^{\mathcal{I}}_{s}$, $\mathcal{D}^{\mathcal{I}}_{t_1} \subset \mathcal{D}^{\mathcal{I}}_{s}$. $\mathcal{D}^{\mathcal{I}}_{t_2}$ partially overlaps $\mathcal{D}^{\mathcal{I}}_{s}$. This can be represented as $\mathcal{D}^{\mathcal{I}}_{s_L} < \mathcal{D}^{\mathcal{I}}_{t_{2L}} < \mathcal{D}^{\mathcal{I}}_{s_R}$ and $\mathcal{D}^{\mathcal{I}}_{s_R} < \mathcal{D}^{\mathcal{I}}_{t_{2R}}$. Where necessary, stage one increases the overall size of the domain interval either by decreasing the left limit or increasing the right limit to reduce the differences. However, in transferring the source to the target, there may be a need to reduce the domain to within the source extremities, either partially or wholly. In Figure 3b, the source domain $\mathcal{D}^{\mathcal{I}}_{s}$, has been reduced to form $\mathcal{D}^{\mathcal{I}}_{t_1}$. $\mathcal{D}^{\mathcal{I}}_{t_2}$ is shifted in a positive direction along the axis. The left limit $\mathcal{D}^{\mathcal{I}}_{t2_L}$ has been moved in a positive direction from $\mathcal{D}^{\mathcal{I}}_{s_L}$. This is accomplished by stage two of the adaptation process. The right limit $\mathcal{D}^{\mathcal{I}}_{t2_R}$ has also been shifted in a positive direction to the outside of the source interval. This is accomplished by stage one.

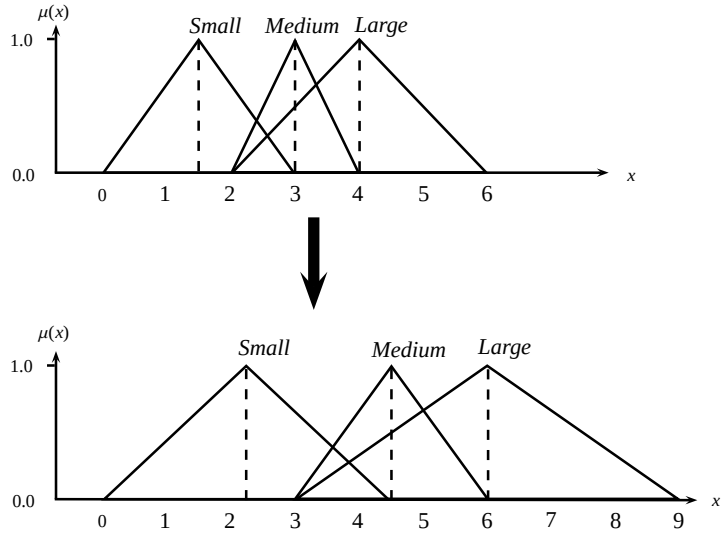
The process used for internal input domain adjustment is illustrated in Figure 4. This diagram shows the flow of the process through three key steps.

- *Step One: Initialisation* This step initialises the system by processing the data from the source domain to gain an input interval. The source domain input interval can be defined as $\mathcal{D}^{\mathcal{I}}_{s}(X) = [x_L, x_R]$ with x_L being the minimum value and x_R being the maximum value of each input value in the domain.

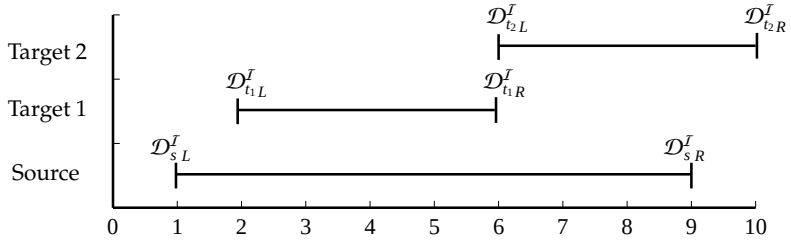
- *Step Two: Correlation* Unlike the source task, the target task has extremely limited data availability. To address this lack of knowledge, the adaptation system uses local minimum and maximum values to compare to the source values. As the target task acquires data points, local minima and maxima are calculated. If one, or both of these values fall within the interval that is represented by the source values x_L and x_R, a proximity measure is produced to ascertain whether the domain is adapted. The proximity is based upon a membership function. The function can take any form chosen, although within the FuzzyTL framework a Gaussian function is used. The membership function is based on the source input domain interval.

  The output from the proximity function is compared to a predefined threshold. When the threshold value is reached, the domains are adapted. The adaptation is based on the values from the target source.

- *Step Three: Negative Influence* Adaptation of the input domains is monitored based on its impact on the overall fuzzy system. To ascertain the influence of this adaptation, a comparison is made between the maximum membership of the rulebase previous to the update, to the same value following the changes. A reduction in



(a) Adaptation of Unevenly Distributed Sets Based on New Minimum and Maximum Input Values.



(b) Example of Internal Domain Containment.

Figure 3: Elements of Adaptive Stages 1 and 2.

value returns the system to its previous state. This allows for the system to police the adaptation, and endeavour to move away from a state of negative transfer.

The first two stages of the adaptation focus on the input variable domains and as a result the antecedent sets. Data is available to produce adaptation within these domains. The unlabelled nature of the data impedes the ability for direct adaptation of the target consequent domains. The third adaptation stage combines data produced by the framework with new task information to approach this problem.

3.3.3. Output Domain Adaptation Through Gradient Control : Stage Three

The third adaptation process focusses on the manipulation of the consequent sets. The process uses information from the target domain coupled with data from the framework itself. This allows for feedback from within the adaptation framework. The process can be summarised as three steps:

- *Step One: Data Gathering* A predefined n sized sliding window $\mathcal{SL}$ of data is collected from the source domain $\mathcal{D}_s$ and the target domain $\mathcal{D}_t$ for each input variable $x \in X$ and output $y \in Y$. The output value for the target domain is taken from the FuzzyTL system. The source output is recorded from the labelled data provided. Gradients are formed based on the sliding window data between the input and the output value. This provides an understanding of the relationship at each data point. The gradients are the basis of the consequent adaptation.

- *Step Two: Gradient Production* Gradients are formed for each source and target input, and the source output. Output from the FuzzyTL framework is used to produce the target output gradient. The data is normalised using

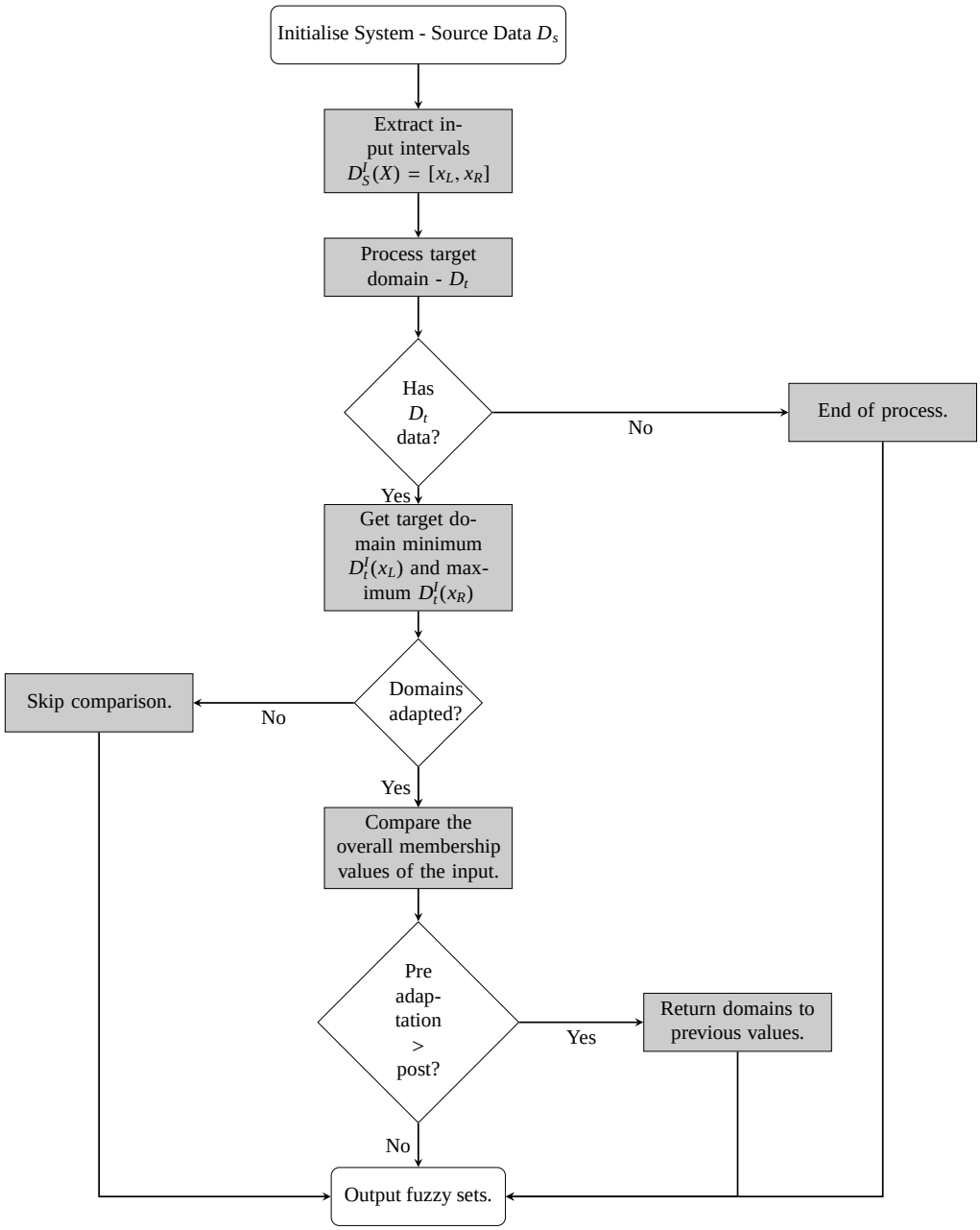


Figure 4: Inner Domain Adaptation Process.

a standard score method. This is defined as:

$$z = \frac{x - \overline{x}}{\sigma} \tag{8}$$

where z is the output, x is the input value, $\overline{x}$ is the mean of the sliding window and σ is the standard deviation of the sliding window.

- *Step Three: Gradient Comparison* Using the gradients gained across each source and target domain input and output variable, a comparison is made at each individual input value.

- *Step Four: Consequent Adaptation* A mapping is made from the source input and output values, to the target input and output values based on the gradients of the values. By mapping the source gradient to the target gradient, differences highlight the necessity to adapt the consequent sets. Differences between the source and target consequent gradients produce adaptations to the target consequent domain interval.

The adaptation of the consequent can be stated as:

$$d\mathcal{D}_a = \varphi \sum_{i=1}^{n} (g_{s_i} - g_{t_i}) \tag{9}$$

where φ is a learning parameter to weight the impact of the gradient delta. This can be user defined although the default is 0.1. g_s is the gradient of the source sliding window for n inputs that can be represented as $g_{s_{i...n}} \in [-1, 1]$,g_t is the gradient of the target sliding window for n inputs that can be represented as $g_{t_{i...n}} \in [-1, 1]$ and $d\mathcal{D}_a$ is the delta used to adapt the consequent sets.

The initial three stages of the adaptation process approach the alteration of the fuzzy sets. In order to absorb the contextual changes within each implementation, the system additionally adapts the fuzzy rulebase.

3.3.4. Rule Base Modification Via Source Rule Comparison : Stage Four

Previous stages focussed on the adaptation of the domains. Differences can also occur within the structure of the rulebase. The knowledge of the FuzzyTL framework is held within the fuzzy sets and fuzzy rules. By altering the rulebase, the knowledge gaps can be filled in order to apply the transferable FIS to the new context.

The rulebase is modified by examining previously pruned rules by applying the target domain data. The use of the exhaustive rulebase is firmly in keeping with the TL ethos of the frameworks construction. Algorithm 1 expresses the adaptation of the rules using the exhaustive transferable rulebase.

The first stage of the process is to examine the exhaustive rulebase $\mathcal{A}$ to identify any rules that fire using the data from the target domain $\mathcal{D}_t$. The rule that fires with the highest membership value from each data point is retained in the adaptive rulebase C. The grouped rules are compared to the reduced rulebase $\mathcal{B}$ based on those with the same antecedent values. Each of the reduced rulebase rules that fires is compared to the adaptive rulebase. Those rules that have greater membership values are retained, removing the comparable rule from the adaptive rulebase. The greater weighting indicates a greater applicability to the new domain. If the identified rule in the reduced rulebase $\mathcal{B}$ is not within the adaptive rulebase C, this is added. The addition of the rules from the exhaustive rulebase assists in supplying missing knowledge areas required by the new task.

The final stage of the adaptation again focuses on the fuzzy rulebase.

3.3.5. Rule Adaptation Using Euclidean Distance Measure : Stage Five

Previously learnt information can provide data to partially fill gaps in the knowledge to complete a new task. To remove incompleteness, and strive to capture all of the segments where disparities lay, new information is required. In the FuzzyTL framework, information of the new domain may only be partially represented in the fuzzy rulebase. To reinforce the rulebase, new rules need to be constructed. As the task domain is an unlabelled dataset, this process relies on the use of the combined learning from the newly accumulated information and the use of previous knowledge in the form of the exhaustive rulebase. To produce antecedent and consequent fuzzy sets, separate strategies are used.

The initial stage of the process is to gain an output from each of the input variables. The process can be demonstrated using a simple example. The domain is segmented into fuzzy sets, $(2N + 1)$. Within this example five sets are used. These are defined as { *Very Low, Low, Medium, High, Very High* }. The input value is applied to each set iteratively. The set with the highest output corresponds to the antecedent set for the new rule. This can be described as:

$$x_1 = \{< \text{Very Low}, 0.0 >< \text{Low}, 0.0 >, < \text{Medium}, 0.35 >, < \text{High}, 0.65 >, < \text{Very High}, 0.0 >\}$$

$$x_2 = \{< \text{Very Low}, 0.0 >< \text{Low}, 0.8 >, < \text{Medium}, 0.2 >, < \text{High}, 0.0 >, < \text{Very High}, 0.0 >\}$$

The example shows two domains, x_1 and x_2 within the target task $\mathcal{D}_t$. The set with the highest firing strength produces the output for the antecedent sets of the rule. If sets of jointly strong firing strength are found, the initially discovered

Algorithm 1 Adaptation Algorithm: Stage Four Adaptation Using Exhaustive Rule Base.

Input Variable $x \in X$
Target Domain $\mathcal{D}_t = \{(x_1^t, x_2^t)\}_t^M$
Exhaustive Rule Base $\mathcal{A} = \{{}^f X_1^a, {}^f X_2^a, {}^f Y^a\}^N$
Reduced Rule Base $\mathcal{B} = \{{}^f X_1^b, {}^f X_2^b, {}^f Y^b\}^P$
Adaptive Rule Base $C = \{{}^f X_1^c, {}^f X_2^c, {}^f Y^c\}^Q$
Membership Degree μ
Rule $r = \{{}^f X_1, {}^f X_2, {}^f Y\}$

for $h = 1; h < M; h++$
  for $i = 1; i < N; i++$
    $e = \mu_{{}_i^{f}X_1^a}({}_h x_1^t) \cdot \mu_{{}_i^{f}X_2^a}({}_h x_2^t)$ ▷ Combine the membership of the antecedent values from the exhaustive rule base.

    if $e > 0$ **then** ▷ Check if the rule produces output.
      for $u = 0 \rightarrow Q; u \leftarrow i + u$
        $g = \mu_{{}_u^{f}X_1^c}({}_h x_1^t) \cdot \mu_{{}_u^{f}X_2^c}({}_h x_2^t)$ ▷ Combine the membership of the antecedent values from the adaptive rule base.

        if $\mu_{{}_i^{f}Y^a}(e) > \mu_{{}_u^{f}Y^c}(g)$ **then** ▷ Compare membership of $\mathcal{A}$ and C rules.
          if $C \neq \emptyset$ **then**
            $C \overset{-}{=} C_k$ ▷ Remove current rule.
          end if
          $C \overset{+}{=} \mathcal{A}_i$ ▷ Add rule from exhaustive rulebase.
        end if
      end for
    end if

    for $j = 1; j < P; j++$
      for $k = 1; k < Q; k++$
        if $({}_j^{f}X_1^b == {}_k^{f}X_1^c) \wedge ({}_j^{f}X_2^b == {}_k^{f}X_2^c) \wedge ({}_j^{f}Y^b == {}_k^{f}Y^c)$ **then**
          ▷ Check if set labels are the same.
          $w = \mu_{{}_u^{f}X_1^b}({}_h x_1^t) \cdot \mu_{{}_u^{f}X_2^b}({}_h x_2^t)$ ▷ Combine the membership of the antecedent values from the reduced rule base.
          if $\mu_{{}_p^{f}Y^b}(w) > \mu_{{}_k^{f}Y^c}(e)$ **then**
            $C \overset{-}{=} C_k$ ▷ Remove current rule.
            $C \overset{+}{=} \mathcal{B}_j$ ▷ Add rule from reduced rulebase.
          end if
        end if
      end for
    end for
  end for
end for

set is used. In the example shown, two antecedent sets relating to two input variables are formed from the highest memberships of the x_1 and x_2 domains. In the x_1 domain, the *High* set produces the highest membership. In x_2, the highest is the set *Low*. As a result the *High* and *Low* are placed into the rule.

The adaptation of the antecedent sets is based on information previously gained. The lack of consequent data requires a different strategy.

There is a mapping between the input values, and the consequent output from a corresponding rule. By using a euclidean distance based on the input values from the unlabelled data, a mapping can be generated to find the closest corresponding consequent set.

During the formation of the exhaustive rulebase, each set is assigned a corresponding input value. Figure 5 shows the relationship between the source and target data values. From the source data, a mapping can be produced from the original input data values to the corresponding antecedent sets. Through the use of an n dimensional euclidean distance, the closet target input values can output sets based on these values. In the example, the sets are represented by the grid structure. Each square shows the relationship of the input values, and the corresponding sets. Shown in Figure 5 are two target antecedent values, t_1 and t_2. These values are not represented within the current reduced rulebase. By mapping them to source input values, the antecedent sets can be found. Using the smallest euclidean value, Figure 5 shows that t_1 can be mapped to s_1 via the distance d_1. Equally, t_2 can be mapped to s_2 via the distance d_4. Through this procedure, a combination of antecedent sets is formed. Using these sets as a comparative value, a consequent set can be extracted from the exhaustive rulebase. By extracting the consequent set in this manner, a new rule is formed that draws knowledge from the source dataset.

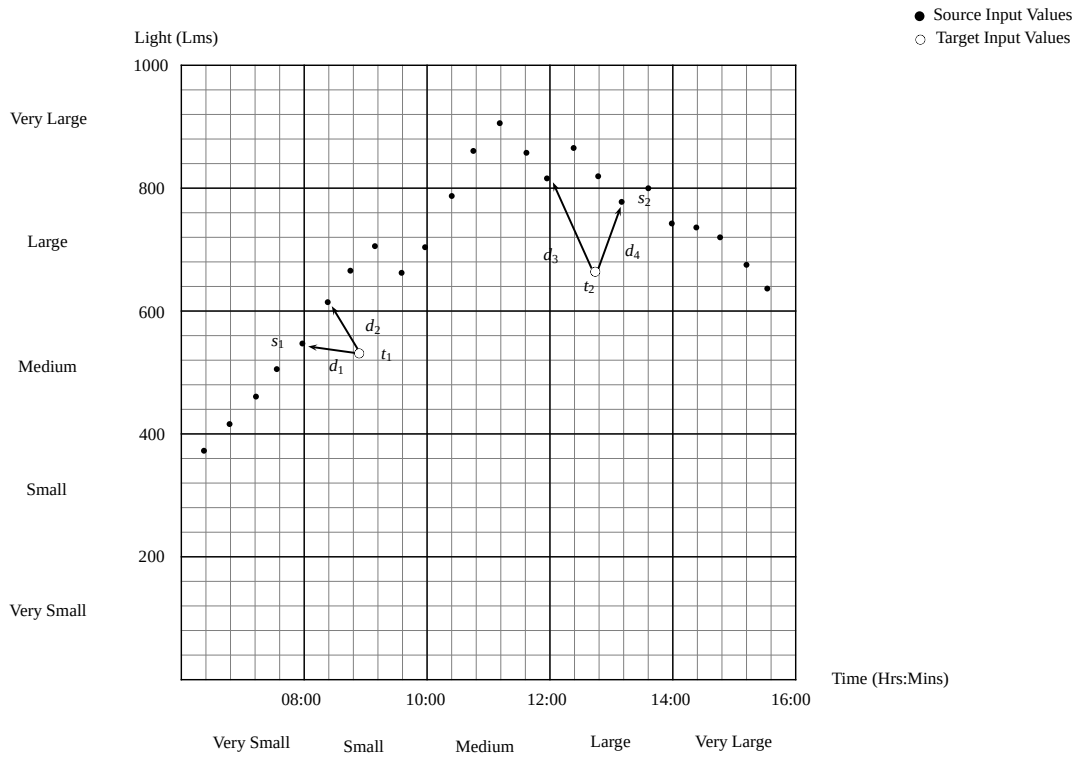


Figure 5: Example of the Gaining of Antecedent Sets to Map Consequent Sets Using Euclidean Distance.

4. Application of Fuzzy Transfer Learning to Intelligent Environments

To illustrate the application of the Fuzzy Transfer Learning methodology, the FuzzyTL was applied to a complex, uncertain and dynamic environment. IEs embody this type of domain. The nature of Intelligent Environment (IE)

applications have produced a number of sparse data problems . Remote locations such as environmental monitoring [37], ad hoc structures within disaster recovery scenarios [38] and highly specific user groups, for example disabled individuals, reduce, or in certain circumstances eliminate, the quantity of data that is available to produce a model.

In this section, the FuzzyTL methodology is shown to be able to predict output from IEs using differing contextual information.

4.1. Experimental Design

The motivation for the FuzzyTL framework is to address the issue of environments where little or no knowledge is known *a-priori*, though there is a need to produce a prediction or classify the target data. Torrey and Shavlik [27] suggest three metrics that can be used to measure the performance of a TL system. These are defined as:

1. The initial performance achievable in the target task using only the transferred knowledge, before any further learning is done, compared to the initial performance of an ignorant agent.
2. The amount of time it takes to fully learn the target task given the transferred knowledge compared to the amount of time to learn it from scratch.
3. The final performance level achievable in the target task compared to the final level without transfer.

These metrics are closely associated with the construction of learning processes, and the composition and quantity of the available data. The metrics proposed by Torrey and Shavlik require the TL structure to be IS, where information is available from both the source and target domains. IU transfer learning, as used in the FuzzyTL framework, makes the comparison of ignorant and informed agents not possible. To compare the performance of a starting ignorant agent is not feasible, as there is no information to model the agent upon. To learn an agent from scratch also requires a level of known data. This research approaches a problem where there is no labelled data within the target domain, and initially little or no unlabelled data. For this reason, the second metric proposed by Torrey and Shavlik is not applicable. On this basis, a different set of metrics are used, however, they are broadly based on the those proposed in [27].

The metrics use a comparison of the output of FuzzyTL framework, against actual known sensor readings from the IEs. Input values are given to the system producing a predictive value. This is compared to actual recorded sensor readings. The error indicates the accuracy of the FuzzyTL system. To evaluate the use of the FuzzyTL framework, two real world IE datasets were chosen to demonstrate the applicability of the spatial and temporal contextual transfer process. The first dataset was taken from a publicly available source [39].

4.1.1. Intel Berkeley Research Laboratory Dataset

The Intel dataset was based upon information collected from 54 sensors deployed in the Intel Berkeley Research Laboratory (hereby referred to as the Intel Laboratory) between the 25th February and the 5th April, 2004. The network used XBow Mica2dot weatherboard based nodes to record environmental data across the internal structure of the laboratory. Four parameters were measured: time-stamped temperature (in degrees Celsius), humidity ranging from 0-100%, light (measured in Lux), and residual power of each sensor unit expressed in Volts. The data was collected using the TinyDB in-network query processing system built onto the TinyOS platform which recorded information every 31 seconds [39].

Additional to the sensor readings, the Intel Laboratory provides the x, y co-ordinates of the sensor locations. These values were used in the experiments to construct a context measure. A section of network was identified to examine the influence of variations in the spatial aspect of the contexts. Data was also taken from a section of the dataset that related to a specific time period. This allowed for the investigation of temporal changes in the context. To achieve both of these, the output of Sensors 7, 9, 12 , 24 , 34 , 42 and 51 where examined across seven days from 28th February to 5th March, 2004 including the 29th February.

A quantity of preprocessing was undertaken to be able to place the dataset into the FuzzyTL framework. Each sensor was isolated based on its moteid. The unused variables were also removed from the data resulting in only the time, light and temperature remaining. The time variable was converted to seconds to allow for ease of processing. The millisecond component was removed allowing for this process. A number of different experimental set ups were used to illustrate the effect of a greater and lesser availability of labelled data in the source domain.

4.2. De Montfort University Robotics Laboratory Dataset

The second dataset is based on a sensor network constructed in the Robotics Laboratory of the Centre for Computational Intelligence of De Montfort University, United Kingdom. Again the sensor network is focussed on the monitoring of environmental conditions. The De Montfort University Robotics Laboratory Sensor Network (here after referred to as the Robotics Laboratory) was composed of nine sensors in total (Sn1 - Sn9). Sensors 3, 5 and 6 (represented as Sn3, Sn5 and Sn6) are composed of Phidget light and temperature sensors. Sensors 1, 2 and 4 (Sn1, Sn2 and Sn4) are single temperature sensors. Five days of data were collected between the 15th and 19th October, 2011. Three of the sensors were used in order to apply the input variables of time and light, and an output variable of temperature. These sensors were Sn3, Sn5 and Sn6. All superfluous information was removed from the raw data leaving each sensor, and the day of the week isolated.

The construction of the framework to process both the Intel Laboratory and Robotics Laboratory datasets were constructed and run using C++ via Code:Blocks (Version 8.02) and compiled through GNU GCC on Ubuntu LTS Version 10.04.

4.2.1. Experiment Structure

To understand the structure of the methodology and its capacity, two hypotheses were proposed and tested.

Hypothesis 1: Where minimal unlabelled data is available within a target task, data in the form of a TL process from contextually related but differing source tasks, can be used to learn predictive tasks.

Hypothesis 2: Adaptation of the transferred source domain through the use of unlabelled new data can increase the performance FuzzyTL in predicting target tasks.

To address these hypotheses, three broad main experimental groups were carried out: performance, context impact and adaptation. Hypothesis 1 was evaluated primarily through the use of the performance and context impact experiments. Hypothesis 2 was evaluated using the adaptation experiments.

Definitions. This section will define a number of elements used through out the remainder of this section. To calculate the difference between the predicted value produced by the FuzzyTL framework and the actual values that are observed, a Root Mean Squared Error (RMSE) is used. The RMSE takes the errors between each of the points in the dataset, and aggregates them into a single measure. RMSE can be defined as

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n} (x_1^i - x_2^i)^2}{n}} \tag{10}$$

were n is the number of data points in the dataset, $x1$ is the observed dataset and $x2$ is the predicted value.

To understand the impact of the contextual change, a metric was produced to measure the temporal-spatial difference between context structure. A normalised euclidean distance was used. Three separate inputs were given, the specification of the sensor location using the x and y co-ordinate, and a date measurement. This is constructed as

$$CD = \sqrt{\frac{((a^1 - a^2) - \sup A)^2}{\inf A - \sup A} + \frac{((b^1 - b^2) - \sup B)^2}{\inf B - \sup B} + \frac{((c^1 - c^2) - \sup C)^2}{\inf C - \sup C}} \tag{11}$$

were a is the x and b is the y co-ordinate, c is the date, and CD is the Context Distance (CD). The supremum and infimum are calculated to produce a normalised distance. This allows for different values to be used. A and B define the spatial inputs of the context. C is the temporal input. The context can be composed of n inputs. The context used in these experiments consists of spatial and temporal elements. Different contexts may contain different quantities of variables.

Context has been defined previously in Section 2.1. Based on this criteria, two further sub-types of context are used, *inter* and *intra*. An inter context can be defined internally within an environment such as section of network or a room in an IE. An intra context is defined as encapsulating the inter context such as building, time scale or composition of these. Within this section, an inter contextual comparison is composed within the individual locations of the sensor networks. Each context is then defined by time and location of the sensors within this location. This falls in line with

the third criteria of a context. An intra contextual comparison uses an abstract definition to compare the contexts. Unlike the inter comparison, a categorical definition is given. These context definitions are used for comparison. Each comparison takes the form of assessing the performance of the FuzzyTL framework based on differences in the individually defined contexts. The difference is calculated based on the defined variables that constituent the context. This allows for contexts with behavioural differences to be compared.

Within the following sections, further explanation of the construction of each individual experiment will be given, with the results gained.

4.2.2. Performance

To measure the performance of the FuzzyTL framework, two datasets were used. Across both datasets, a system was constructed using two input variables (time and light), and a single output variable, temperature. For the performance measure, inter contextual comparisons were produced. The initial value output of the system was based on zero prior unlabelled data. Each data point from the dataset was fed into the system to simulate real time operation. To assess the performance of the FuzzyTL framework the predicted value at each data point was compared to the actual observed output from the dataset. Any error produced was consolidated into a single value using the RMSE process.

To produce a benchmark to compare the system against, each of the datasets were processed using the adapted Fuzzy Frequency WM system [36]. The learning process was, however, altered. The source data was supplied from the target domain. This produced an output that, unlike the FuzzyTL, has labelled knowledge of the target learning task. This allows a comparison to be made. The FuzzyTL framework that was supplied target data perceived to produce an output closest to the actual sensor value. A comparison was made to demonstrate the context of the output from the contextually different source FuzzyTL framework.

Intel Laboratory Data Comparison to Observed Values. For the Intel Laboratory dataset, individual contexts were formed using each of the sensors in the spatial grouping (Sensors 7, 9, 12 , 24 , 34 , 42 and 51), and for each day between 28th February to 5th March, 2004. Extraneous source and target contexts were removed from the experiment. A single context is formed from a single day and a single sensor. A value was predicted for a single sensor taken from within the group, across the defined time period. This produced 2352 differing context comparisons. In the event that the system is unable to predict a reading, a predefined value of -1 is given.

To assess the performance of the FuzzyTL framework, source contexts were compared to the benchmark values. The adapted WM method produced 49 separate RMSE values (seven sensors $\times$ seven days). These related to each sensor (7, 9, 12 , 24 , 34 , 42 and 51), and each day within the specified time interval (from 28th February to 5th March, 2004 including the 29th February). 2352 contexts were produced for the Intel Laboratory comparison. These were composed of the 49 separate contexts from the source data (seven sensors $\times$ seven days) and the target data (seven sensors $\times$ seven days) with 49 contexts removed where the source and target context matched. From the 2352 RMSE values calculated from the FuzzyTL framework, the lowest RMSE value was taken for each context. This produced 49 contexts to compare to the benchmark set. By isolating the lowest RMSE values, this equated to the *best* performing contexts. As the benchmark dataset represents the optimum data conditions for the learning process, the best performing contexts were chosen to compare against them. The focus of this process is to establish whether the FuzzyTL framework can firstly output a predictive value, and to then contextualise the performance.

The data for the experiment was composed of two sample population datasets, the benchmark and Intel Laboratory datasets. To compare the benchmark of the Intel Laboratory Data and the output of the FuzzyTL, the datasets was normalised using a power transform, $log_6(x)$.

To compare the two sample populations, a paired t-test was used. The paired t-test produced an value of 0.0015. As this value was below the α of 0.05 it was concluded that the benchmark and FuzzyTL datasets were from non-identical populations. Looking closer at the data, 37 contexts were highlighted as producing a negative difference. This showed that the best performing FuzzyTL output was greater in 37 contexts than the benchmark. However, 12 of the contexts, 24.4898%, produced a lower RMSE than the benchmark. In these cases the FuzzyTL system was able to use contextually different source data to produce better performing output than the benchmark dataset.

Figure 6 shows a single context comparison where the best FuzzyTL output out performed the benchmark dataset. This is the source data 24, 28th February, 2004, and the target data Sensor 34, 28th February, 2004.

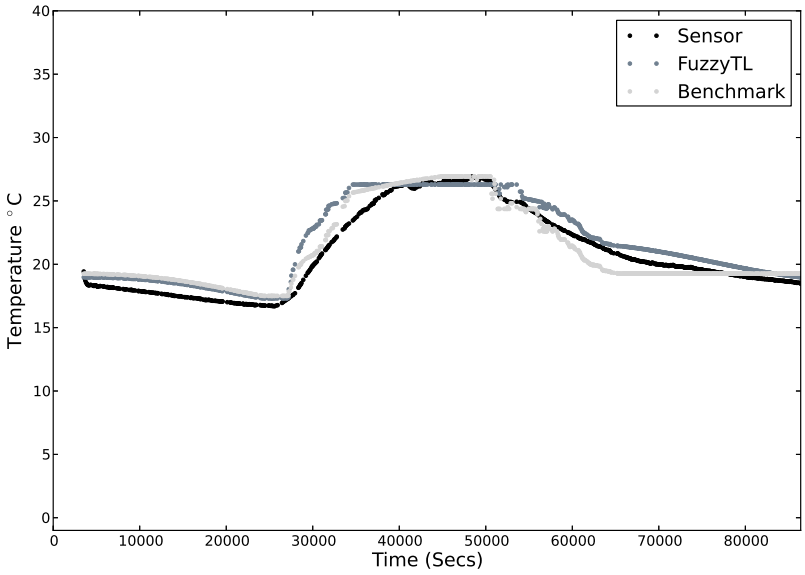


Figure 6: Comparison of Benchmark and Best FuzzyTL to *Sensor Readings Target Data Sensor 34, 28th February, 2004*.

The ability for the framework to adapt the sets according to new data improved the output beyond the benchmark. The final values of the input domains were in close proximity to the target sensor as is expected. The output domain had an increased difference. The left temperature interval value (tm_L) differed by 0.18°C, and the right temperature interval value differed by 0.86°C. As the system is dynamic, this is absorbed within the process.

Of the 2352 contexts, 66.1990% (1557 out of 2352) produced a RMSE that was equal, or within the minimum and maximum interval of the benchmark dataset. This indicates that the FuzzyTL was able to use differing contextual source data to produce comparable predictive output. The lowest of those contexts produced a RMSE of 0.5139°C. The data used was source data from sensor 42 on the 3rd March, 2004 and target data from the sensor 7 on the 2nd March, 2004. In comparison, the highest error produced was a RMSE of 10.5014°C. The source data was provided by sensor 7 on the 4th March, 2004, and the target data by sensor 24 on the 2nd March, 2004. The returned RMSE can be attributed to the nature of the target data. The different structure of the interval domains of the source and target consequent sets, produced a variation in the output compared to the actual value.

The highest source consequent had a left limit value y_L^{s1} of 17.2640°C. The right limit value y_R^{s1} was 36.1584°C. The target interval was an intersection of this interval. The left target limit value y_L^t was 14.3044°C and the right target limit was 21.2624°C. In contrast, the lowest error rate had a closer consequent domain interval. The source had a left limit y_L^{s2} of 16.9308°C and a right limit y_R^{s2} of 31.5034°C. The target had a left limit of 17.8520°C and a right limit of 24.8100°C. The knowledge in the consequent domain is dependent on the source task. The closer the source and target consequent domain intervals, the smaller the RMSE that is produced.

Robotics Laboratory Data Comparison to Observed Values. A similar analysis was undertaken using the Robotics Laboratory dataset. The adapted WM methodology produced 12 RMSE values (four days × three sensors) based on the sensors in the Robotics laboratory and across the defined number of days. These values were compared to the lowest, and so best, output produced by the FuzzyTL framework using different contextual data.

As with the Intel dataset, a paired t-test was carried out between the Robotics Laboratory dataset and the output of the FuzzyTL. The p-value that was produced was lower than the defined α significance value of 0.05. This rejects the null hypothesis that the Robotics Laboratory benchmark dataset is the same distribution as the best output of the FuzzyTL framework. Further analysis showed that all contexts produced a higher RMSE value than the benchmark.

Focussing closer on the data showed that individual source contexts produced RMSE values that were comparable to the benchmark and similar to the actual sensor output. Overall, the lowest RMSE value produced was using source data from sensor 1 on the 18th October 2011, to predict values for sensor 1 on 17th October 2011. The RMSE value for the context was 0.1862°C. The benchmark produced RMSE values in the range of a RMSE of 0.0987 to 0.2212°C. The RMSE for this context is 0.0987°C. The Robotics laboratory produces a more consistent temperature than found within readings recorded in the Intel dataset. The variation across the context shown was 0.67°C. The FuzzyTL framework replicates the narrow variation producing a value of 0.55°C. The initial value calculated, based on zero data, produced an error of only 0.11°C.

The highest RMSE produced was a value of 3.3447, using source data from sensor 2 on the 16th October, 2011, to predict values for sensor 1 on 17th October, 2011. The variation across both the output of the sensor and predictive value is again low, 0.89 and 0.83°C respectively. The impact of the size of the output interval will be discussed in the following section.

5. Adaptation

The FuzzyTL framework is grounded on the transfer and adaptation of information. To investigate the performance gain through the use of the adaptation process, a comparison was made between a non-adapted system, and the full FuzzyTL framework.

5.1. Comparison of Non-Adaptive FuzzyTL to Full FuzzyTL Framework: Intel Laboratory Data

The non-adaptive system was composed of a transferred FIS. The learning processes involved in forming the structure of the fuzzy system remained unchanged to those previously used. Supplementary online adaptation and learning was removed providing a base to compare to. The first experiment was based on the Intel laboratory dataset. The structure outlined in Section 4.1.1 was used to form the basis for the comparison. Using a similar performance metric, each of the different contexts were compared to the sensor readings. The non-adaptive RMSE values were subsequently compared to the values previously gained from using the full FuzzyTL system. A total of 2352 contexts were used for comparison.

The previous experiments focussed on samples of the datasets. This experiment used the whole set of both non-adapted and adapted data. To compare the two datasets, a Wilcoxon signed-rank test was chosen. The p-value for the Wilcoxon signed-rank test was lower than the defined alpha value of 0.05. This indicated that the two datasets were from different distributions. Further analysis showed that 87.6700% (2062 of 2352) of contexts exhibited a decrease in RMSE when the adaptation was applied.

Isolating a single context for comparison, Figure 7 shows the comparison of the non-adaptive and full FuzzyTL systems using source data from sensor 7 on 2nd March, 2004, and target data from sensor 24 on 2nd March, 2004. This illustrates the non-adaptive systems inability to cope with the initial prediction, producing a -1 value. This is due to the nature of the input, and output domain intervals. Issues arise within the non-adaptive system as the input values fall outside of the light interval domain. The minimum of the target light interval (l_L^{ts}) sits outside of the non-adapted light interval (l_L^{na}), $l_L^{na} > l_L^{ts}$. The $l_L^{ts} = 97.52$ and $l_L^{na} = 158.23$. The results of this are shown in Figure 7 between 51 to 27260 seconds (00:00:51 - 07:34:20). The non-adaptive system can not produce an output based on these inputs.

Taking on board the transferred information, and the new data, the FuzzyTL framework adapts the all interval domains. Driven by the inputs from both the target and source data, the output interval domain is shown, in this example, to move toward the parameters of the sensor interval. The minimum and the maximum FuzzyTL light values (l_L^{ftl} and l_R^{ftl}) are equal actual sensor. The output values, are are additionally closer to the actual sensor interval minimum and maximum values, than the non-adapted system. Overall, the incorporation of the adaptation stages into the FuzzyTL improved the predictive capability when used with the Intel Laboratory dataset.

5.2. Comparison of Non-Adaptive FuzzyTL to Full FuzzyTL Framework: Robotics Laboratory Data

In a similar approach to the Intel laboratory dataset, a comparison was made between a non-adaptive and full FuzzyTL system using the Robotics laboratory data. The non-adaptive system was based on the FuzzyTL framework with the removal of the adaptive stages.

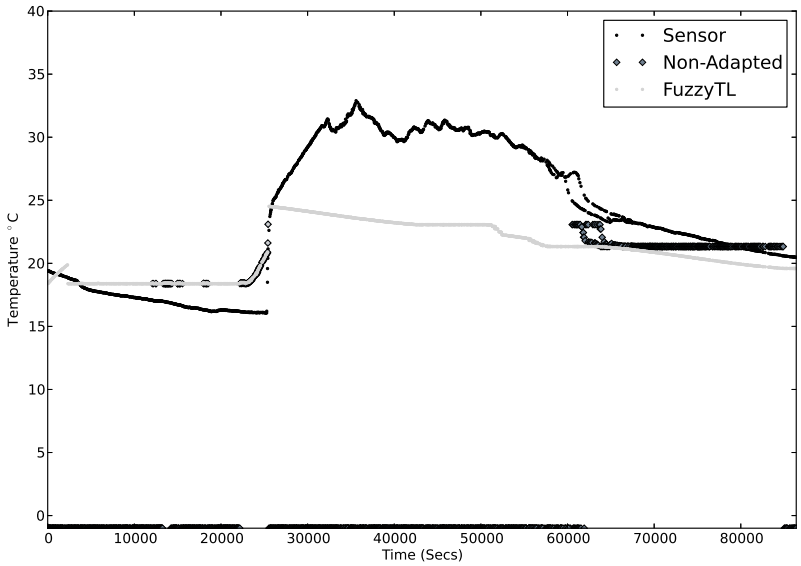


Figure 7: Comparison of Non-Adapted System, FuzzyTL and Sensor *Source Data Sensor 7, 2nd March, 2004 and Target Data Sensor 24, 2nd March, 2004* based on the Intel Laboratory Dataset.

To compare the data, a Wilcoxon signed-rank test was used. The p-value calculated was lower than the alpha value of 0.05. Taking the null hypothesis stated that both datasets come from the same distribution, this can be rejected.

The differences between the datasets showed that 82.5757% (109 of 132) of the contexts produced a lower RMSE when the adaptive stages were applied. Of those, the greatest reduction was an RMSE of 14.2349°C. By comparison, of the 17.4242% that produced a higher RMSE, the greatest increase was 1.1318°C. These findings substantiate the previous conclusions drawn from the Intel Laboratory dataset.

Drilling down into the contexts, Figure 8 shows an example of non-adapted system output compared to the actual sensor reading. This figure shows how, in certain conditions, the non-adapted system is unable to output a value for a portion of the target (Within Figure 8, the missing output is not displayed).

As the light input value steps outside of the domain, the system fails to produce an output. The transferred light domain is $l_L = -0.01$ and $l_R = 811.01$. At 36842 seconds (10:14:02) the input light level reaches 863, beyond the light domain interval. This causes the system to fail to output a value.

5.3. Comparison to Other Regression Methods

To demonstrate the effective nature of FuzzyTL, two other regression based methods were chosen to compare the approach to. The authors chose to compare the FuzzyTL method to an implementation of K-Nearest Neighbour (KNN) and Support Vector Regression (SVR). These methods were chosen as they come from differing areas of machine learning which are mature and established methodologies. Each method was applied to the Intel dataset as defined in Section 4.1.1. The implementations of both KNN and SVR used in this paper are taken from Pedregosa et al's [**?**] Python based data mining and analysis tool set, Scikit-learn.

5.3.1. Implementation of K-Nearest Neighbour

The use of the KNN method for regression has been applied to a number of applications from predicting log volumes [**?**] to uses with large data sources in astronomy [**?**].

KNN implements learning based on the k nearest neighbours of each query point, where k is an integer value specified by the user. The target is predicted by local interpolation through the use of the targets associated with

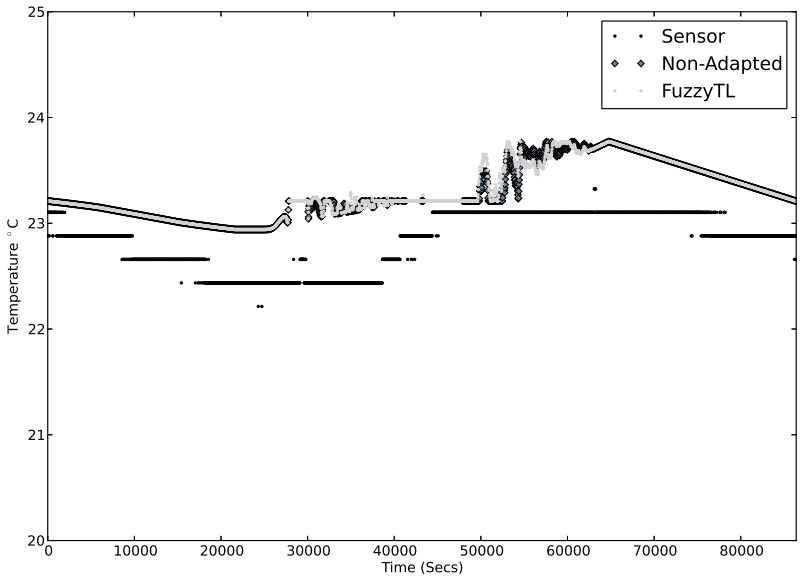


Figure 8: Comparison of Non-Adapted System, FuzzyTL and Sensor *Source Data Sensor 2, 16th October, 2011 and Target Data Sensor 3, 17th October, 2004* based on the Robotics Laboratory Dataset.

the nearest neighbour in the training set. In this case, a value of five was specified. For standard KNN, each point in the local neighbourhood contributes equally to the classification of the query point. The implementation used in this paper, however, alternatively weighted the points such that nearby points contribute more to the regression than faraway points. This gives greater emphasis to the data closer to the current information. The Scikit-learn implementation achieves this through weights proportional to the inverse of the distance from the query point [**?**]. The sparse nature of the data forced the use of a brute-force search to locate the neighbours in the training data. Although inefficient in large datasets, the nature of the training data lends itself to this approach.

5.3.2. Implementation of Support Vector Regression

The use of Support Vector Machine (SVM) and additionally the use of SVR has become increasingly wide spread. They have successfully been applied to many applications that require classification, for example text classification [**?**], gene selection for Cancer classification [**?**] and gesture detection [**?**]. The derivative of SVM, SVR, has been equally applied to real-world problems. SVR has been applied to applications ranging from face detection [**?**] and financial forecasting [**?**], to affective computing [**?**].

SVM is a method that was developed at AT&T Bell laboratories by Vapnik and a number of co-workers during the 1990's (Boser, Guyon and Vapnik 1992, Guyon, Boser and Vapnik 1993,Cortes and Vapnik, 1995, Scholköpf, Burges and Vapnik 1995, 1996, Vapnik, Golowich and Smola 1997 [**?**]). In 1996 Drucker et al. [**?**] introduced a regression extension of the SVM. In simple terms, SVR looks to find a function, $f(x)$, with at most ϵ-deviation from the target y based on training data $(x_1, y_1) \ldots, (x_l, y_l) \in X$ where X denotes the space of the input patterns. The vector is required to be as flat as possible. Errors are accepted as long as they are less than ϵ but no deviation can be greater. A detailed examination and explanation of the methodology of SVR is beyond the scope of this paper. For a greater insight into SVR, see [**?**].

A number of parameters were set to define the SVR algorithm within the Scikit-learn implementation. There is a large body of research into the tuning of these hyperparameters [40]. A discussion of tuning these parameters, is again beyond the scope of this research. As a result a generalised approach was taken to the parameter structure. A C value of 1.0 was set as the penalty of the error term. A value of 0.1 was specified for the epsilon-tube. The kernel function

was defined as a Guassian Radial Basis Function. This is a popular kernel method [40] for SVR implementations. Other parameters used the default settings.

5.3.3. Results

The comparisons to both the KNN and SVR methodologies used the same dataset as discussed in Section 4.2.2. 2352 comparisons were made using the Intel Laboratory dataset. The results illustrated that the FuzzyTL produced a lower RMSE value in 54.5493% (1283 out of 2352) of the comparisons made with the KNN method and 67.8996% (1597 out of 2352) compared to the SVR approach. Looking closer at the results, it can be seen that in a high proportion, the performance of the FuzzyTL approach was near comparable to both the KNN and SVR. In an additional 26.4856% of the comparisons for the KNN method and 11.1630% in the case of SVR, the FuzzyTL methodology was within 0.5% RMSE. This highlights that even in those contexts that the FuzzyTL framework produced a greater overall error, the margin was comparable to the performance of the other methods.

6. Conclusion

The following sections discuss the findings of the experiments and offer some discussion surrounding the key outcomes.

6.1. Contextually Differing Environments Can Act as Source Information

A main focus of this paper was to understand whether the FuzzyTL framework could produce a predictive output based on little knowledge of the target domain. To asses the frameworks output, a comparison was made against two benchmark datasets using a RMSE value. The Intel Laboratory dataset comparison showed that in 24.4898% of cases, the FuzzyTL produced a lower RMSE value than the benchmark. Additionally, of the Intel Laboratory contexts analysed (2352 in total), 66.1990% contextually different source datasets produced an RMSE output that was equal, or within the minimum or maximum interval of the benchmark dataset. The analysis highlighted that the FuzzyTL framework was able to produce output that matched or surpassed the defined benchmark.

Overall, the Robotics Laboratory dataset substantiated the Intel Laboratory findings. Again a paired t-test was carried out to compare the best FuzzyTL output to the benchmark results. Unlike the Intel Laboratory comparison, all of the contexts studied produced a higher RMSE when using contextually different source data. Drilling down into the data, however, showed that individual contextual instances produced RMSE values that were comparable to the benchmark.

The FuzzyTL was shown to be able to output predictive values using contextually different source data. Output from the FuzzyTL framework was comparable to a benchmark formed using target information. Although stronger within the Intel Laboratory dataset, the FuzzyTL framework was shown to produce predictive output across two differing real-world datasets.

6.2. Online Adaptation Decreases the Error of the FuzzyTL Output

To investigate the impact of the adaptation process, and test the second hypothesis, a series of experiments based on a non-adapted version of the FuzzyTL framework were used. The Non-Adaptive (N-A) framework was constructed from a transferred FIS.

A comparison was made between the performance of the N-A framework and the full FuzzyTL framework. The output of each context was assessed against the actual sensor readings. Of the 2352 contexts compared, 2062 (87.6700%) produced a lower RMSE when the adaptive stages were incorporated. This clearly demonstrated that the introduction of the adaptive stages decreased the error produced. Failures in the N-A framework produced high RMSE values. The strict structure of the N-A framework resulted in failures. Target values that were beyond the source domain failed to produce an output. The adaptive process allowed the FuzzyTL to alter the domains to the target data, producing an output. In a minority of specific cases, the N-A frameworks out performed the full FuzzyTL framework. These special cases required the source and target interval domains to be in close proximity. Additionally, the target input domains were required to be proper subsets of the source input domains.

The same comparison was carried out on the Robotics Laboratory dataset. Of the differences defined, 82.5757% (109 of the 132 contexts) produced a lower RMSE when the adaptive process was applied. Across the contexts, the largest reduction was 14.2349°C.

6.3. FuzzyTL can outperform other regression based algorithms using contextually different sparse data

A comparison was made between the FuzzyTL framework and two mature algorithms, KNN and SVR. 2352 contextually differing comparisons were made using the Intel Laboratory dataset. Predictive values were produced from each different methodology and compared to the actual output from the real-world data. When compared to the SVR approach, the FuzzyTL framework produced a lower RMSE in 1597 out of 2352 (67.8996%) contexts. The same comparison process was used between FuzzyTL and a KNN implementation. Of the 2352 context, the FuzzyTL framework produced a lower RMSE in 1283 (54.5493%).

Further analysis showed that the FuzzyTL framework was able to outperform the other methodologies in contexts were significant adaptation was required.

7. Future Work

A number of areas of the Fuzzy Transfer Learning can be expanded.

7.1. Comparison of Wang-Mendel Method to Other Rule Generation Methodologies

Within the FuzzyTL framework, a WM methodology was used to extract a FIS using numerical data. This research carried out a preliminary comparison of FIS production methods. Further methods exist outside of this study, though the comparison of these was outside the scope of this paper. An in depth comparison of extraction methodologies would highlight other applicable methodologies. This may allow for the further extension of the FuzzyTL framework.

7.2. Automation of Set Extraction From Data

The current FuzzyTL framework uses a simple system to define the number of sets that are used within the FIS. This is based on a preassigned value. Previous research has been carried out into the extraction of fuzzy sets from labelled data. Of interest is the application of such methods to automate the process of assigning set quantities. The removal of the need to use expert knowledge to assign set quantities would further automate the framework. This area of research would extend the data driven nature of the framework structure.

7.3. Use of Multiple Context Data

The composition of the source data has been shown to have a direct effect on the outcome of the predictive value of the FuzzyTL framework. The proximity of the source and target domains can have a direct effect on the error produced. Certain conditions of a source domain can have adverse affects on the learning of the target domain. Anomalous or erroneous data can produce a model that is incorrect. To tackle this issue, research has been conducted into the use of multiple source domains within TL [41, 42]. The use of multiple sources can increase the chance of discovering a source domain that is close to the target. The extension of the FuzzyTL framework to incorporate multiple source data may reduce the impact of negative transfer.

References

[1] W. Dai, Y. Chen, G. Xue, Q. Yang, Y. Yu, NIPS 2008 (2008) 353–360.
[2] H. Larochelle, D. Erhan, Y. Bengio, in: AAAI Conference on Artificial Intelligence, volume 1, pp. 2–2.
[3] O. Chapelle, B. Schölkopf, A. Zien, et al., Semi-supervised learning, volume 2, MIT press Cambridge, MA:, 2006.
[4] D. Perkins, G. Salomon, International encyclopedia of education 2 (1992) 1.
[5] S. Thrun, in: Advances in Neural Information Processing Systems, pp. 640–646.
[6] Q. Xu, Q. Yang, Journal of Computing Science and Engineering 5 (2011) 257–268.
[7] N. Gorski, J. Laird, in: Proceedings of the ICML-06 Workshop on Structural Knowledge Transfer for Machine Learning.
[8] D. Hu, Q. Yang, in: Twenty-Second International Joint Conference on Artificial Intelligence, pp. 1962–1967.
[9] S. Barrett, M. Taylor, P. Stone, in: Ninth International Conference on Autonomous Agents and Multiagent Systems-Adaptive Learning Agents Workshop (AAMAS-ALA).
[10] J. Mendel, Proceedings of the IEEE 83 (1995) 345–377.
[11] L. Zadeh, Computer 21 (1988) 83–93.
[12] A. Dey, Personal and ubiquitous computing 5 (2001) 4–7.
[13] P. Dourish, Personal Ubiquitous Comput. 8 (2004) 19–30.
[14] B. Schilit, N. Adams, R. Want, in: Proceedings of the 1994 First Workshop on Mobile Computing Systems and Applications-Volume 00, pp. 85–90.

[15] S. Jang, Gwangju Institute of Science and Technology, PhD thesis (2005).
[16] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, D. Riboni, Pervasive and Mobile Computing 6 (2010) 161–180.
[17] L. Zadeh, Software, IEEE 11 (1994) 48 –56.
[18] S. Guillaume, Fuzzy Systems, IEEE Transactions on 9 (2001) 426 –443.
[19] J. Mendel, Signal Processing 80 (2000) 913–933.
[20] L. Wang, J. Mendel, Systems, Man and Cybernetics, IEEE Transactions on 22 (1992) 1414–1427.
[21] J. Casillas, O. Cordón, F. Herrera, in: Proceedings of the 8th Information Processing and Management of Uncertainty in Knowledge-Based Systems Conference, pp. 1682–1688.
[22] L. Mihalkova, T. Huynh, R. Mooney, in: Proceedings of the national conference on artificial intelligence, volume 22, p. 608.
[23] J. Shell, S. Vickers, S. Coupland, H. Istance, in: Computational Intelligence (UKCI), 2012 12th UK Workshop on, pp. 1–8.
[24] M. Sharma, M. Holmes, J. Santamaria, A. Irani, C. Isbell, A. Ram, in: Proceedings of the Twentieth International Joint Conference on Artificial Intelligence, 1041-1046.
[25] A. Quattoni, M. Collins, T. Darrell, in: Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on, IEEE, pp. 1–8.
[26] S. Pan, Q. Yang, IEEE Transactions on Knowledge and Data Engineering (2009) 1345–1359.
[27] L. Torrey, J. Shavlik, Handbook of Research on Machine Learning Applications. IGI Global 3 (2009) 17–35.
[28] D. Cook, K. Feuz, N. Krishnan, Transfer learning for activity recognition: A survey, http://www.eecs.wsu.edu/ cook/pubs/kais12.pdf, 2012. Accessed 15th November, 2012.
[29] E. Miller, Learning from one example in machine vision by sharing probability densities, Ph.D. thesis, Yale University, 2002.
[30] J. Shell, S. Coupland, in: Intelligent Environments (IE), 2011 7th International Conference on, pp. 149–156.
[31] A. Sengupta, T. Pal, European Journal of Operational Research 127 (2000) 28–43.
[32] R. Moore, Methods and applications of interval analysis, volume 2, Society for Industrial Mathematics, Philadelophia., 1987.
[33] D. Teodorovic, P. Lucic, J. Popovic, S. Kikuchi, B. Stanic, in: Fuzzy Systems, 2001. The 10th IEEE International Conference on, volume 1, pp. 276–279.
[34] X. Yang, J. Yuan, J. Yuan, H. Mao, Expert Systems with Applications 37 (2010) 8036–8041.
[35] F. Doctor, H. Hagras, V. Callaghan, IEEE Transactions on Systems, Man, and Cybernetics, Part A: Systems and Humans 35 (2005) 55–65.
[36] J. Shell, S. Coupland, Ambient Intelligence (2012) 145–160.
[37] G. Werner-Allen, K. Lorincz, M. Ruiz, O. Marcillo, J. Johnson, J. Lees, M. Welsh, Internet Computing, IEEE 10 (2006) 18–25.
[38] G. Zussman, A. Segall, Ad Hoc Networks 1 (2003) 405–421.
[39] S. Madden, Intel lab data, http://db.csail.mit.edu/labdata/labdata.html, 2004. Published on 2nd June 2004.
[40] K. Duan, S. S. Keerthi, A. N. Poo, Neurocomputing 51 (2003) 41–59.
[41] P. Luo, F. Zhuang, H. Xiong, Y. Xiong, Q. He, in: Proceedings of the 17th ACM conference on Information and knowledge management, CIKM '08, ACM, New York, NY, USA, 2008, pp. 103–112.
[42] Y. Yao, G. Doretto, in: Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on, pp. 1855–1862.

Jethro Shell has over 10 years experience in industry working for Panasonic Computer Products (Europe). Whilst in industry he studied for his Masters in Information Technology at De Monfort University, Leicester, U.K., graduating in 2008. He left industry to pursue a career in academia, gaining his Ph.D in computer science in 2013. Whilst studying he received the British Computer Society "Current Post-graduate Research in Computing" award for his work on transfer learning and fuzzy logic. He is currently a research fellow with the DIGITS research group at De Montfort University, Leicester, U.K.

Simon Coupland received the B.Sc. (Hons.) degree in computing from University College Northampton, Northampton, U.K., in 2002 and the Ph.D. degree from De Montfort University, Leicester, U.K., in 2006, which was funded by the U.K. Engineering and Physical Sciences Research Council.

He is a Senior Research Fellow with the Centre for Computational Intelligence, De Montfort University, where he is involved in the field of type-2 fuzzy logic. His research interests include the theory and application of generalized type-2 fuzzy logic.

Dr. Coupland was the first author of the paper which received the 2007 IEEE TRANSACTIONS ON FUZZY SYSTEMS Outstanding Paper Award and was the joint winner of the 2008 British Computer Society Machine Intelligence Award.